



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/790,302	03/01/2004	Michael David Marr	MSFT-3031/306162.01	9280
41505	7590	08/24/2007	EXAMINER	
WOODCOCK WASHBURN LLP (MICROSOFT CORPORATION)			CHEN, QING	
CIRA CENTRE, 12TH FLOOR			ART UNIT	PAPER NUMBER
2929 ARCH STREET			2191	
PHILADELPHIA, PA 19104-2891			MAIL DATE	DELIVERY MODE
			08/24/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)
	10/790,302	MARR ET AL.
	Examiner	Art Unit
	Qing Chen	2191

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --
Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1) Responsive to communication(s) filed on 01 March 2004.
 2a) This action is **FINAL**. 2b) This action is non-final.
 3) Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4) Claim(s) 1-25 is/are pending in the application.
 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
 5) Claim(s) _____ is/are allowed.
 6) Claim(s) 1-25 is/are rejected.
 7) Claim(s) _____ is/are objected to.
 8) Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9) The specification is objected to by the Examiner.
 10) The drawing(s) filed on 01 March 2004 is/are: a) accepted or b) objected to by the Examiner.
 Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
 Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
 11) The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12) Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
 a) All b) Some * c) None of:
 1. Certified copies of the priority documents have been received.
 2. Certified copies of the priority documents have been received in Application No. _____.
 3. Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) Notice of References Cited (PTO-892)
 2) Notice of Draftsperson's Patent Drawing Review (PTO-948)
 3) Information Disclosure Statement(s) (PTO/SB/08)
 Paper No(s)/Mail Date 20040301.

4) Interview Summary (PTO-413)
 Paper No(s)/Mail Date. _____.
 5) Notice of Informal Patent Application
 6) Other: _____.

DETAILED ACTION

1. This is the initial Office action based on the application filed on March 1, 2004.
2. **Claims 1-25** are pending.

Oath/Declaration

3. The oath or declaration is defective. A new oath or declaration in compliance with 37 CFR 1.67(a) identifying this application by application number and filing date is required. See MPEP §§ 602.01 and 602.02.

The oath or declaration is defective because:

- It does not identify the city and either state or foreign country of residence of each inventor. The residence information may be provided on either an application data sheet or supplemental oath or declaration.
- It does not fully identify the mailing address of each inventor. A mailing address is an address at which an inventor customarily receives his or her mail and may be either a home or business address. The mailing address should include the ZIP Code designation. The mailing address may be provided in an application data sheet or a supplemental oath or declaration. See 37 CFR 1.63(c) and 37 CFR 1.76.

Drawings

4. The drawings are objected to as failing to comply with 37 CFR 1.84(p)(5) because they include the following reference character(s) not mentioned in the description:

- Reference number 708 in Figure 7.

Corrected drawing sheets in compliance with 37 CFR 1.121(d), or amendment to the specification to add the reference character(s) in the description in compliance with 37 CFR 1.121(b) are required in reply to the Office action to avoid abandonment of the application.

Any amended replacement drawing sheet should include all of the figures appearing on the immediate prior version of the sheet, even if only one figure is being amended. The figure or figure number of an amended drawing should not be labeled as "amended." If a drawing figure is to be canceled, the appropriate figure must be removed from the replacement sheet, and where necessary, the remaining figures must be renumbered and appropriate changes made to the brief description of the several views of the drawings for consistency. Additional replacement sheets may be necessary to show the renumbering of the remaining figures. Each drawing sheet submitted after the filing date of an application must be labeled in the top margin as either "Replacement Sheet" or "New Sheet" pursuant to 37 CFR 1.121(d). If the changes are not accepted by the Examiner, the Applicant will be notified and informed of any required corrective action in the next Office action. The objection to the drawings will not be held in abeyance.

Specification

5. The disclosure is objected to because of the following informalities:
 - Reference number 136(2) should read -- 135(2) -- on page 10, paragraph [0037].
 - "At the time that program module 136 is called from 206" should presumably read -- At the time that program module 136 is called from *application 135* -- on page 11, paragraph [0040].Appropriate correction is required.
6. The use of trademarks, such as WINDOWS, has been noted in this application. Trademarks should be capitalized wherever they appear (capitalize each letter OR accompany

each trademark with an appropriate designation symbol, *e.g.*, TM or [®]) and be accompanied by the generic terminology (use trademarks as adjectives modifying a descriptive noun, *e.g.*, “the WINDOWS operating system”).

Although the use of trademarks is permissible in patent applications, the proprietary nature of the marks should be respected and every effort made to prevent their use in any manner, which might adversely affect their validity as trademarks.

Claim Objections

7. **Claims 1, 2, 5, 8-17, and 24** are objected to because of the following informalities:

- **Claim 1** contains a typographical error: “permitted to *invoked* said functionality” should read -- permitted to *invoke* said functionality --.
- **Claim 2** recites the limitation “the first module.” Applicant is advised to change this limitation to read “the first software module” for the purpose of providing it with proper explicit antecedent basis.
- **Claim 5** contains a typographical error: “to cause a return to the location *to* which a return should have occurred” should presumably read -- to cause a return to the location *in* which a return should have occurred --.
- **Claims 8 and 9** recite the limitation “said first program module.” Applicant is advised to change this limitation to read “said first software module” for the purpose of providing it with proper explicit antecedent basis.

- **Claims 10 and 12** recite the limitation “said first module.” Applicant is advised to change this limitation to read “said first software module” for the purpose of providing it with proper explicit antecedent basis.
- **Claims 13-17** depend on Claim 12 and, therefore, suffer the same deficiency as Claim 12.
- **Claim 11** recites the limitation “said second program module.” Applicant is advised to change this limitation to read “said second software module” for the purpose of providing it with proper explicit antecedent basis.
- **Claim 12** contains the following typographical errors:
 - “permitted” should read -- permitting --.
 - “to identify a return address *to* which control of the process will return” should presumably read -- to identify a return address *in* which control of the process will return --.
- **Claim 24** contains a typographical error: the word “and” should be added after the first limitation.

Appropriate correction is required.

Claim Rejections - 35 USC § 112

8. The following is a quotation of the second paragraph of 35 U.S.C. 112:

The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

9. **Claims 5, 13, and 22-25** are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

Claim 5 recites the limitation “the next occurring return instruction.” There is insufficient antecedent basis for this limitation in the claim. In the interest of compact prosecution, the Examiner subsequently interprets this limitation as reading “a next occurring return instruction” for the purpose of further examination.

Claim 13 recites the limitation “said act of permitting execution of said second program module to proceed.” There is insufficient antecedent basis for this limitation in the claim. In the interest of compact prosecution, the Examiner subsequently interprets this limitation as reading “said act of permitting execution of said first program module to proceed” for the purpose of further examination.

Claims 22 and 24 recite the limitations “the return address” and “the next return instruction.” There are insufficient antecedent bases for these limitations in the claims. In the interest of compact prosecution, the Examiner subsequently interprets these limitations as

reading “a return address” and “a next return instruction,” respectively, for the purpose of further examination.

Claim 25 depends on Claim 24 and, therefore, suffers the same deficiencies as Claim 24.

Claim 23 recites the limitation “said return address.” There is insufficient antecedent basis for this limitation in the claim. In the interest of compact prosecution, the Examiner subsequently interprets this limitation as reading “a return address” for the purpose of further examination.

Claim 24 recites the limitation “said second value.” There is insufficient antecedent basis for this limitation in the claim. In the interest of compact prosecution, the Examiner subsequently interprets this limitation as reading “a second value” for the purpose of further examination.

Claim 25 depends on Claim 24 and, therefore, suffers the same deficiency as Claim 24.

Claim 24 recites the limitation “computer-readable medium comprising computer-executable instructions.” The claim is rendered indefinite because a computer-readable medium cannot comprise computer-executable instructions. Computer instructions are stored on a computer-readable medium so as to become structurally and functionally interrelated to the computer-readable medium. In the interest of compact prosecution, the Examiner subsequently

interprets this limitation as reading “computer-readable medium storing computer-executable instructions” for the purpose of further examination.

Claim 25 depends on Claim 24 and, therefore, suffers the same deficiency as Claim 24.

Claim Rejections - 35 USC § 101

10. 35 U.S.C. 101 reads as follows:

Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

11. **Claims 18-25** are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

Claims 18-23 are directed to program modules. However, the recited components of the systems appear to lack the necessary physical components (hardware) to constitute a machine or manufacture under § 101. Therefore, these claim limitations can be reasonably interpreted as computer program modules—software *per se*. The claims are directed to functional descriptive material *per se*, and hence non-statutory.

The claims constitute computer programs representing computer listings *per se*. Such descriptions or expressions of the programs are not physical “things.” They are neither computer components nor statutory processes, as they are not “acts” being performed. Such claimed computer programs do not define any structural and functional interrelationships between the computer program and other claimed elements of a computer, which permit the computer

program's functionality to be realized. In contrast, a claimed computer-readable medium encoded with a computer program is a computer element, which defines structural and functional interrelationships between the computer program and the rest of the computer, that permits the computer program's functionality to be realized, and is thus statutory. See *Lowry*, 32 F.3d at 1583-84, 32 USPQ2d at 1035.

Claims 24 and 25 recite computer-readable medium as a claimed element. However, it is noted that the specification describes such computer-readable medium as comprising communication media, which typically embodies computer-readable instructions, data structures, program modules or other data in a modulated data signal such as a carrier wave or other transport mechanism and includes any information delivery media (see *Page 6, Paragraph [0025]*). Consequently, the computer-readable medium can be reasonably interpreted as carrying electrical signals.

Claims that recite nothing but the physical characteristics of a form of energy, such as a frequency, voltage, or the strength of a magnetic field, define energy or magnetism *per se*, and as such are non-statutory natural phenomena. *O'Reilly v. Morse*, 56 U.S. (15 How.) 62, 112-14 (1853). Moreover, it does not appear that a claim reciting a signal encoded with functional descriptive material falls within any of the categories of patentable subject matter set forth in § 101.

Claim Rejections - 35 USC § 102

12. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless --

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

13. **Claims 1-8, 10-12, and 16-25** are rejected under 35 U.S.C. 102(b) as being anticipated by Pekowski (US 5,946,486).

As per **Claim 1**, Pekowski discloses:

- from the first software module, issuing a call to a first method that invokes a functionality performed by the second software module (*see Figure 3; Column 4: 67 through Column 5: 1-3, "... the shadow DLL acts as an interceptor of all calls being made to the target DLL, and thus each call to the target DLL and return from the target DLL passes through the shadow DLL."*);

- verifying that the call originated from a source that is permitted to invoke said functionality (*see Column 9: 40-43, "At the shadow DLL's entry point function, the stack contains the caller's return address 600 and the caller's parameter 610 pushed onto the stack by the caller (see FIG. 5A)."*);

- performing said functionality (*see Figure 3; Column 5: 37-39, "When called, the shadow DLL performs the entry hook processing and performs the call on the original DLL."*); and

- returning to said first software module (*see Figure 3; Column 5: 39-41, "When the original DLL returns from the call, the shadow DLL performs the exit hook processing and returns to the original caller."*).

As per **Claim 2**, the rejection of **Claim 1** is incorporated; and Pekowski further discloses:

- wherein said first method is performed by the second software module, said first method being exposed to the first software module, said first method performing said functionality (*see Figure 3; Column 4: 67 through Column 5: 1-3, "... the shadow DLL acts as an interceptor of all calls being made to the target DLL, and thus each call to the target DLL and return from the target DLL passes through the shadow DLL."*).

As per **Claim 3**, the rejection of **Claim 1** is incorporated; and Pekowski further discloses:

- wherein said first method is performed by a third software module that invokes said functionality in the second software module (*see Figure 3; Column 4: 67 through Column 5: 1-3, "... the shadow DLL acts as an interceptor of all calls being made to the target DLL, and thus each call to the target DLL and return from the target DLL passes through the shadow DLL."*).

As per **Claim 4**, the rejection of **Claim 3** is incorporated; and Pekowski further discloses:

- wherein said call is made according to a first calling convention, and wherein said third software module uses a second calling convention different from said first calling convention to invoke said functionality in the second software module (*see Column 9: 43-47, "At the shadow DLL's common entry function, the stack contains the caller's return address 600,*

the caller's parameters 610, the first time called flag 602, entry only flag 604, hook value 606, and target DLL's entry point address 608." and 52-56, "At the target DLL's entry point function, the stack contains the caller's parameters 610 pushed onto the stack by the caller and the return address 630 of element allocated from the return function table replaced by the common entry function (see FIG. 5C).").

As per **Claim 5**, the rejection of **Claim 4** is incorporated; and Pekowski further discloses:

- wherein said second calling convention causes a program stack to be modified in order to cause a next occurring return instruction to cause a return to the location in which a return would have occurred if a return had been executed following a call to said third software module and prior to a call to said second software module (see Column 9: 52-56, "At the target DLL's entry point function, the stack contains the caller's parameters 610 pushed onto the stack by the caller and the return address 630 of element allocated from the return function table replaced by the common entry function (see FIG. 5C).").

As per **Claim 6**, the rejection of **Claim 3** is incorporated; and Pekowski further discloses:

- wherein said first calling convention comprises placing a first return address on a stack, said first return address representing a location at which execution is to resume after a function call is completed, and wherein said second calling convention comprises placing a second return address on a stack and placing data at the location represented by said second return address (see Column 9: 43-47, "At the shadow DLL's common entry function, the stack contains the caller's return address 600, the caller's parameters 610, the first time called flag

602, entry only flag 604, hook value 606, and target DLL's entry point address 608." and 52-56, "At the target DLL's entry point function, the stack contains the caller's parameters 610 pushed onto the stack by the caller and the return address 630 of element allocated from the return function table replaced by the common entry function (see FIG. 5C)."), and wherein the method further comprises:

- using said second return address to find said data, said data being used for at least one of:
 - performing said verifying act; and
 - identifying a location to execute in order to perform said functionality (see

Column 9: 52-56, "At the target DLL's entry point function, the stack contains the caller's parameters 610 pushed onto the stack by the caller and the return address 630 of element allocated from the return function table replaced by the common entry function (see FIG. 5C).").

As per **Claim 7**, the rejection of **Claim 1** is incorporated; and Pekowski further discloses:

- examining a call stack to identify a return address, and determining that the return address is part of a program module that is permitted, according to a standard or rule, to invoke said functionality (see *Column 9: 40-43, "At the shadow DLL's entry point function, the stack contains the caller's return address 600 and the caller's parameter 610 pushed onto the stack by the caller (see FIG. 5A).")*.

As per **Claim 8**, the rejection of **Claim 7** is incorporated; and Pekowski further discloses:

- determining that said return address is from a location or range of locations within said first software module from which invocation of said functionality is permitted to originate (see Column 9: 40-43, “At the shadow DLL's entry point function, the stack contains the caller's return address 600 and the caller's parameter 610 pushed onto the stack by the caller (see FIG. 5A).”).

As per **Claim 10**, the rejection of **Claim 1** is incorporated; and Pekowski further discloses:

- wherein said first software module is, or is part of, an application program (see *Figure 3: 150*).

As per **Claim 11**, the rejection of **Claim 1** is incorporated; and Pekowski further discloses:

- wherein said second software module comprises a dynamic-link library (see *Figure 3: 160*).

As per **Claim 12**, Pekowski discloses:

- examining a call stack of a process in which said first program module executes to identify a return address in which control of the process will return upon completion of a call to said first program module (see Column 9: 52-56, “At the target DLL's entry point function, the stack contains the caller's parameters 610 pushed onto the stack by the caller and the return

address 630 of element allocated from the return function table replaced by the common entry function (see FIG. 5C).");

- determining that said return address is located within a second program module that is permitted to call said first software module (*see Column 9: 52-56, "At the target DLL's entry point function, the stack contains the caller's parameters 610 pushed onto the stack by the caller and the return address 630 of element allocated from the return function table replaced by the common entry function (see FIG. 5C)."); and*
- based on the result of said determining act, permitting execution of said first program module to proceed (*see Figure 3; Column 5: 37-39, "When called, the shadow DLL performs the entry hook processing and performs the call on the original DLL. ".*).

As per **Claim 16**, the rejection of **Claim 12** is incorporated; and Pekowski further discloses:

- wherein said first program module is called by a third program module, said third program module having a callable method exposed to said second program module, said callable method causing said first program module to be invoked and passing said return address to said first program module at the time that said first program module is invoked (*see Figure 3; Column 4: 67 through Column 5: 1-3, "... the shadow DLL acts as an interceptor of all calls being made to the target DLL, and thus each call to the target DLL and return from the target DLL passes through the shadow DLL. ".*).

As per **Claim 17**, the rejection of **Claim 16** is incorporated; and Pekowski further discloses:

- wherein said first program module adjusts the content of said call stack to reflect that said first program module will return to said return address upon completion of execution of said call to said first program module, said call stack reflecting, in the absence of the adjustment, that said first program module will return to an address other than said return address (see Column 9: 52-56, “*At the target DLL's entry point function, the stack contains the caller's parameters 610 pushed onto the stack by the caller and the return address 630 of element allocated from the return function table replaced by the common entry function (see FIG. 5C). ”*).

As per **Claim 18**, Pekowski discloses:

- a function that is performable on behalf of a calling entity (see Figure 3; Column 5: 37-39, “*When called, the shadow DLL performs the entry hook processing and performs the call on the original DLL. ”*); and

- logic that verifies an identity of the calling entity as a condition for performing said function, said logic consulting a call stack in order to identify said calling entity (see Column 9: 40-43, “*At the shadow DLL's entry point function, the stack contains the caller's return address 600 and the caller's parameter 610 pushed onto the stack by the caller (see FIG. 5A). ”*).

As per **Claim 19**, the rejection of **Claim 18** is incorporated; and Pekowski further discloses:

- wherein said logic determines said identity based on a return address on said call stack, said return address representing a location of an instruction to be executed when the program module completes execution (*see Column 9: 40-43, "At the shadow DLL's entry point function, the stack contains the caller's return address 600 and the caller's parameter 610 pushed onto the stack by the caller (see FIG. 5A). "*).

As per **Claim 20**, the rejection of **Claim 18** is incorporated; and Pekowski further discloses:

- wherein said function is not exposed to said calling entity, and wherein said function is exposed to an intermediate entity that is callable by said calling entity, said intermediate entity calling upon the program module to perform said function on behalf of said calling entity (*see Figure 3; Column 4: 67 through Column 5: 1-3, "... the shadow DLL acts as an interceptor of all calls being made to the target DLL, and thus each call to the target DLL and return from the target DLL passes through the shadow DLL. "*).

As per **Claim 21**, the rejection of **Claim 20** is incorporated; and Pekowski further discloses:

- wherein said calling entity calls said intermediate entity using a first calling convention, and wherein said intermediate entity calls the program module using a second calling convention different from said first calling convention (*see Column 9: 43-47, "At the shadow DLL's common entry function, the stack contains the caller's return address 600, the caller's parameters 610, the first time called flag 602, entry only flag 604, hook value 606, and*

target DLL's entry point address 608." and 52-56, "At the target DLL's entry point function, the stack contains the caller's parameters 610 pushed onto the stack by the caller and the return address 630 of element allocated from the return function table replaced by the common entry function (see FIG. 5C).").

As per **Claim 22**, the rejection of **Claim 21** is incorporated; and Pekowski further discloses:

- wherein said calling entity calls said intermediate entity by calling a function in said intermediate entity and placing a first return address on said call stack, and wherein said intermediate entity calls the program module by calling or jumping to a location in said program module with one or more parameters including said first return address, wherein the program module verifies that said first return address is located within a calling entity that is allowed to call the program module, and wherein the program module adjusts said call stack so that a return address to be followed upon a next return instruction is equal to said first return address (see *Column 9: 43-47, "At the shadow DLL's common entry function, the stack contains the caller's return address 600, the caller's parameters 610, the first time called flag 602, entry only flag 604, hook value 606, and target DLL's entry point address 608.").*

As per **Claim 23**, the rejection of **Claim 21** is incorporated; and Pekowski further discloses:

- wherein said first calling convention comprises placing a return address on said call stack, and wherein said second calling convention comprises placing a second return address on

said call stack and placing data at the location represented by said second return address, said second return address being used to find said data, said data being used to perform at least one of: verification of the identity of the calling entity; and identification of a location at which said function is located (*see Column 9: 52-56, "At the target DLL's entry point function, the stack contains the caller's parameters 610 pushed onto the stack by the caller and the return address 630 of element allocated from the return function table replaced by the common entry function (see FIG. 5C). "*).

As per **Claim 24**, Pekowski discloses:

- receiving a first call or first jump from a first entity, there being a call stack which, at the time of said first call or first jump, has a state in which a return address to be executed upon a next return instruction is equal to a first value (*see Figure 3; Column 4: 67 through Column 5: 1-3, "... the shadow DLL acts as an interceptor of all calls being made to the target DLL, and thus each call to the target DLL and return from the target DLL passes through the shadow DLL. "; Column 9: 43-47, "At the shadow DLL's common entry function, the stack contains the caller's return address 600, the caller's parameters 610, the first time called flag 602, entry only flag 604, hook value 606, and target DLL's entry point address 608. "); and*
- issuing a second call or a second jump to a second entity, the second call or second jump being parameterized by one or more values including said first value, said second entity having access to a second value and using a second value to verify which return address was applicable at the time that said first call or said first jump was made, said second entity adjusting said call stack to set a return address equal to the first value (*see Column 9: 52-56, "At the target*

DLL's entry point function, the stack contains the caller's parameters 610 pushed onto the stack by the caller and the return address 630 of element allocated from the return function table replaced by the common entry function (see FIG. 5C).").

As per **Claim 25**, the rejection of **Claim 24** is incorporated; and Pekowski further discloses:

- wherein said second entity returns directly to said first entity without returning to an intermediate entity that received said first call or said first jump (see *Figure 3; Column 5: 39-41, "When the original DLL returns from the call, the shadow DLL performs the exit hook processing and returns to the original caller. "*).

Claim Rejections - 35 USC § 103

14. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

15. **Claims 9 and 13** are rejected under 35 U.S.C. 103(a) as being unpatentable over Pekowski (US 5,946,486) in view of Cronce (US 6,880,149).

As per **Claim 9**, the rejection of **Claim 1** is incorporated; however, Pekowski do not disclose:

- verifying that said first software module, or a portion thereof, has not been modified relative to a previously-known state.

Cronce discloses:

- verifying that said first software module, or a portion thereof, has not been modified relative to a previously-known state (*see Column 2: 3-14, "The resulting executable code is delivered as a protected software application that generates a new checksum at runtime and compares it with the computed checksum, and determines that the software program has been modified if the checksums fail to match."*).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Cronce into the teaching of Pekowski to include verifying that said first software module, or a portion thereof, has not been modified relative to a previously-known state. The modification would be obvious because one of ordinary skill in the art would be motivated to validate code base integrity (*see Cronce – Column 2: 15-17*).

As per **Claim 13**, the rejection of **Claim 12** is incorporated; however, Pekowski do not disclose:

- determining that said second program module, or a portion thereof, has not been modified relative to a previously-known state of said second program module, wherein said act of permitting execution of said first program module to proceed is further based on the determination as to whether said second program module has been modified.

Cronce discloses:

- determining that said second program module, or a portion thereof, has not been modified relative to a previously-known state of said second program module, wherein said act of permitting execution of said first program module to proceed is further based on the determination as to whether said second program module has been modified (*see Column 2: 3-14, "The resulting executable code is delivered as a protected software application that generates a new checksum at runtime and compares it with the computed checksum, and determines that the software program has been modified if the checksums fail to match. "*;

Column 3: 19-21, "If the checksums compare, then the code is not changed, and execution begins of the main application code 100. ".).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Cronce into the teaching of Pekowski to include determining that said second program module, or a portion thereof, has not been modified relative to a previously-known state of said second program module, wherein said act of permitting execution of said first program module to proceed is further based on the determination as to whether said second program module has been modified. The modification would be obvious because one of ordinary skill in the art would be motivated to validate code base integrity (*see Cronce – Column 2: 15-17*).

16. **Claims 14 and 15** are rejected under 35 U.S.C. 103(a) as being unpatentable over Pekowski (US 5,946,486) in view of Downs et al. (US 6,226,618).

As per **Claim 14**, the rejection of **Claim 12** is incorporated; however, Pekowski do not disclose:

- wherein said first program module includes logic that resists misuse and/or tampering with said first program module, and wherein said determining act is performed by said first program module.

Downs et al. disclose:

- wherein said first program module includes logic that resists misuse and/or tampering with said first program module, and wherein said determining act is performed by said first program module (*see Column 80: 60-64, "IBM's tamper-resistant software made it difficult to circumvent these copy protection mechanisms. This is a very typical application for tamper-resistant software; the software is used to enforce rules on the usage of some protected type of Content 113. "*).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Downs et al. into the teaching of Pekowski to include wherein said first program module includes logic that resists misuse and/or tampering with said first program module, and wherein said determining act is performed by said first program module. The modification would be obvious because one of ordinary skill in the art would be motivated to deter unauthorized entry into a computer software application by a hacker (*see Downs et al. – Column 80: 41-43*

As per **Claim 15**, the rejection of **Claim 12** is incorporated; however, Pekowski do not disclose:

- wherein said first program module comprises cryptographic functionality that stores and obscures a decryption key and that uses said decryption key to decrypt content, or uses said decryption key as part of a process of decrypting content.

Downs et al. disclose:

- wherein said first program module comprises cryptographic functionality that stores and obscures a decryption key and that uses said decryption key to decrypt content, or uses said decryption key as part of a process of decrypting content (*see Column 10: 61-65, "Once these verifications are satisfied, the Clearinghouse(s) 105 sends the decryption key for the Content 113 to the requesting End-User(s) packed in a License SC. The key is encrypted in a manner so that only the authorized user can retrieve it."*).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Downs et al. into the teaching of Pekowski to include wherein said first program module comprises cryptographic functionality that stores and obscures a decryption key and that uses said decryption key to decrypt content, or uses said decryption key as part of a process of decrypting content. The modification would be obvious because one of ordinary skill in the art would be motivated to allow only the authorized user to have access to the secured content (*see Downs et al. – Column 10: 41-43*).

Conclusion

17. The prior art made of record and not relied upon is considered pertinent to Applicant's disclosure.

Any inquiry concerning this communication or earlier communications from the Examiner should be directed to Qing Chen whose telephone number is 571-270-1071. The Examiner can normally be reached on Monday through Thursday from 7:30 AM to 4:00 PM. The Examiner can also be reached on alternate Fridays.

If attempts to reach the Examiner by telephone are unsuccessful, the Examiner's supervisor, Wei Zhen, can be reached on 571-272-3708. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the TC 2100 Group receptionist whose telephone number is 571-272-2100.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

WZ ✓
WEI ZHEN
SUPERVISORY PATENT EXAMINER

QC / ac
July 31, 2007